LA-UR -83-3394

CONF-831082--4

Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36

TITLE DATABASE TRANSFERS BETWEEN SEVERAL SYSTEMS

AUTHOR(S) Margaret Doll

SUBMITTED TO Paper to be used in the Proceedings of the
1983 Fall DECUS U.S. Symposium
Las Vegas, NV
October 24-28, 1983

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Los Alamos Los Alamos National Laboratory
Los Alamos, New Mexico 87545

DATABASE TRANSFERS BETWEEN SEVERAL SYSTEMS

Margaret Doll[*]
Los Alamos National Laboratory
Los Alamos, New Mexico 87545

ABSTRACT

The ability to transfer databases between systems allows the user to exploit the best features of either system. This paper addresses beginning Datatrieve users and deals with the issues involved in a transfer of a database from a central computing area to a PDP-11 at the Los Alamos National Laboratory. FRAMIS was used to clean the original database; DATATRIEVE was used to establish the new database. The new database residing on the PDP-11 was subject to structural change at any time.

INTRODUCTION

The purpose of this paper is to show transfer techniques of datasets to a PDP-11 and development of a Datatrieve dataset. The original database dealt with in this paper contained information identifying the user, his terminal's location, a location code for the mainframe and jackfields through which his communication line ran, and the port number into the concentrator. Our group's responsibility for the communication line ended at this point. The original database contained records with varying definitions.

Following are terms that will be used throughout the paper.

Database: One dataset or a group of datasets that contain information concerning a common subject. The group of datasets has at least one cross reference.

Dataset: A file or records where all records are concerned with a common aspect of a subject, where each record contains information concerning a unique entry and where the file has a definition of fields or attributes across each of the records or tuples.

Channel: A unique communication line.

Terminal Port: A port into a switch, concentrator or multiplexer that ends a channel.

NAMES.DAT: Early in this paper, NAMES.DAT will refer to a set of five destination datasets differentiated by security partitions in which the channel entries belonged. Later the five sets were combined into a single dataset. Channel technical and administrative information is contained in NAMES.DAT.

PATHS.DAT: Early in this paper, PATHS.DAT will refer to a set of five destination datasets differentiated by security partitions in which the channel entries belonged. This group of datasets is no longer being used at Los Alamos National Laboratory. The channels path with all the connected peripherals were to be contained in PATHS.DAT.

See Fig. 1 for an illustration of the following terms.

Mainframe: A physical frame where wires from the user's side of a channel are connected to wires from a concentrator or multiplexer. Mainframes are used to physically separate wires into groups by security partition.

Jackfield: A physical frame where wires from the user's side of a channel to a concentrator or multiplexer can be spliced into testing equipment.

Concentrator: An interface that checks passwords, checks security compatibility between user and destination computer and packages and unpackages signals between the user and the destination computer.

Multiplexer: A destination computer that checks passwords, checks security compatibility between user and itself and serves many users at one time.

[*] *This work performed under the auspices of the U. S. Department of Energy contract W-7405-ENG-36.*

THE NEED FOR A NEW DATABASE

It was necessary to do a major overhaul to the original database in order to regain the integrity of the data, to gain an interactive database and to allow a channel line-testing program to link with the database (see Fig. 2). The integrity of the old database had been corrupted.

(1) Entries into the database had been made through an editor instead of a database program. Therefore, data for corresponding data fields or attributes did not always begin in the same column on each record.

(2) Data had been entered by only one data entry person, who though familiar with the coding through years of experience, had no experience in the engineering aspect of a communication line. Therefore this person could not check the data thoroughly nor make suggestions as to additional data that might be useful in day to day operations.

(3) After the initial recording of data, there was no updating procedure in place. Data was changed only when the user or the terminal port was changed.

There was a need for an interactive database.

(1) Billing information had to be maintained as before. However, inventory reports were needed for each of our user groups in order to verify our records and to let each group be more aware of their resources.

(2) Instantaneous information concerning communication channels was needed by the technicians on our Trouble Desk. The technicians had been using a written listing of the information in the old database. We wanted programs that could access up-to-date information, as well as a database that would contain more information about cross connections, etc.

(3) Designs for cuts of new channels did not always relate to the actual wiring available. A way was necessary to inventory channels not connected to a port, a user, and channels that were in use.

Originally, the plan was to use a PDP-11 to drive test equipment on communication lines as trouble calls came in and as part of a routine procedure. The PDP-11 would have to contain the database with complete channel information. This plan has been laid aside while our new database is being developed.

CLEANING THE OLD DATABASE

The old database was cleaned in the following ways using FRAMIS (see Fig. 3):

(1) All records that did not have standardized information were thrown away.

(2) All records that were too incomplete to have any value were eliminated.

(3) Consistent typing errors were corrected.

(4) Duplicates were eliminated.

(5) Coding was standardized.

(6) Only data fields that had relevant data and data that seemed to be up-to-date were selected.

(7) Records having information on the same communication channel were put together.

(8) And blank fields whose values could be determined were filled in.

For an illustration of the "cleaned" channel database see Fig. 4.

Our new database was to consist of two datasets for each of the four security partitions in our Laboratory with two datasets for incomplete channels. Therefore, the "cleaned" data was split into five datasets before the transfer.

THE TRANSFER

The transfer of data from our Common File System was done through a VAX. The stored files in standard text were translated into VAX native text: then transferred to tape using FLEX (see Fig. 5)

CREATION OF NAMES.DAT AND PATHS.DAT

The remaining transfer commands were contained in one command file on the PDP-11 (see Figs. 6-8).

Data from tape for the five files was copied on the PDP-11 using FLEX (see Fig. 6).

Files on the PDP-11 were modified through EDT to change the lowercase character VAX files to uppercase files. One by one the data files were entered into the EDT editor, e.g., EDT ANAMES.DAT. NOKEYPAD was set and the editor was changed from line mode to screen mode. CHGCER tells the editor to change character case from the current pointer's location to the end of range. In this case the pointer was at the beginning of the file. End-of-range was the end of the file. The first "EX" exited the editor from screen mode back to line mode. The second "EX" exited from the editor, saving the new file. The call to the EDT editor could be written into a command file, but I found no way to include commands to EDT in a command file.

The tape files consisted of a set of three consecutive records. Tapeform, a simple Fortran program combined each set of records into one record (see Fig. 7). Five new datasets NNAMES.DAT;1 through NNAMES.DAT;5 were created

DATATRIEVE was called to establish destination data files and dataset definitions (see Fig. 8 for the command file calling DATATRIEVE and Figs 9 and 12 for the file definitions). The names of the fields and the type of characters to be used in the fields were established. The indexes of the file and the original allocations for the files were created.

The next step was to create a dataset definition over the files that had been processed. The command file FNAMES.CMD defined the domain NNAMES over NNAMES.DAT;*, the files created by TAPEFORM (see Fig. 10).

FNAMES.CMD also transferred data from NNAMES.DAT;5 to FNAMES.DAT using a routine similar to Fig. 11 where ANAMES is appropriately changed to FNAMES.

It was not necessary to spell out the storage as explicitly as was done in Fig. 11.

(1) CONTACT-LAST and CONTACT-FIRST could have been transferred by the statement CONTACT=CONTACT.

(2) The transfer would have been faster if more group fields had been designed into the dataset definition. For instance SP, STATUS, XMSN, FORMAT, SPEED, SIGNALLING, and TYPE could have grouped under one main heading.

(3) And if the transferred records had been arranged exactly as the destination datasets were to be, there would have been no need for the storage by DATATRIEVE at all. Rather, the dataset definitions could have been defined over the files coming from TAPEFORM. Unfortunately, at the time of the data transfer, the destination dataset definitions were changing daily. The use of DATATRIEVE procedures to reshuffle data took less programming time than to rewrite Fortran programs or to restart the data transfer for the original database in the Common File System.

Originally NAMES.DAT was to contain billing information and user terminal information such as speed, signalling, etc. along channels. PATHS.DAT would contain wiring information (see Fig. 12).

Channels in PATHS.DAT would begin at SE-NO 1 (for the user end of the channel) or at SEQ-NO 31500 (for a channel beginning at a switch). The channel would proceed along various routes to a final SEQ-NO of 32000 (for a port on a concentrator or multiplexor) or SEQ-NO of 31000 (for a port on a switch). (See Fig. 13)

ULOCATN.CMD,...,ALOCATN.CMD transferred data from NAMES.DAT concerning the beginning of the channels (see Figs. 14 and 15).

TAG, STATUS, BEG-BOX and BEG-LOC were transferred from NAMES.DAT to a temporary file defined in Fig. 14. TAG was the cross index between files. STATUS indicated whether the channel was a maintenance channel, privately assigned, not being used, etc. BEG-BOX contained port numbers for the channels beginning at SEQ-NO 31500; these channels did not begin with a user (see Fig. 13, Channel No. 3). BEG-LOC contained the area, building and room in which the user's terminal was located.

Data from records of channels beginning with a user was transferred to PATHS.DAT using the procedures in the central area of Fig. 15.

Location had been entered into the original database in various formats. Thus it was necessary to redefine this field in three ways in the temporary dataset before its transfer to PATHS.DAT (see Fig. 14). Fortunately, the three variations included nearly all of the forms in this field, and the subfields consistently divided by slashes in the original database. DATATRIEVE tested to see where the slashes divided the subfields in records with user information in BEG-LOC and stored the data into PATHS.DAT accordingly.

Information from records that pertained to channels beginning with a switch port were transferred to SEQ-NO 31500 records in PATHS.DAT (see Fig. 15 at the bottom of the page). The beginning location was provided by the DATATRIEVE procedure.

The NPATHS.CMD files created ending records describing the end of the channel in the PATHS.DAT datasets (see Fig. 16). Usually the ending of a channel was a port into a concentrator or multiplexer. We designated concentrator and multiplexer records as SEQ-NO 32000. However, some channels ended in a port switch line side, "PSL" (see Fig. 13, Channel No. 2). The corresponding SEQ-NO was 31500. HWA.DEV was the destination variable for the terminating port on the concentrator, multiplexer or switch (see Fig. 16).

CHANGES TO THE DESTINATION DATABASE

Since the database was created, the following changes have been made (see Fig. 17):

(1) PATHS.DAT was eliminated. Maintenance of all datasets and the effort to provide procedures for data entrance and reports from the beginning of the transfer proved too much of an effort. Also, smaller wiring datasets seem to be a better investment rather than the all encompassing datasets PATHS.DAT. The smaller datasets will be datasets on data communications equipment, jackfields, etc. These datasets can be maintained through DATATRIEVE procedures by the engineers who design the cuts for the channels. The smaller datasets can have definitions that suit their specific needs and can be maintained by experts on the information the datasets contain. NAMES.DAT will be maintained by one person in charge of the billing. However, it is necessary to have one database manager to maintain cross-reference indexes and standard coding of common information across the datasets.

(2) We found datasets separated by security partitions was not necessary and, in fact, it was a hindrance. For example, when the Trouble Desk received a call on a channel, it required less time and effort to search one dataset instead of four.

(3) Field lengths changed as we gained experience on the information needed in the datasets.

(4) Fields were renamed and query names were added to shorten typing efforts.

(5) Certain fields were grouped together for easier access for reports or transfers to other datasets.

(6) New fields were added.

(7) And data was reshuffled. For example the location field in NAMES.DAT has been divided into three fields for easier searches on ports used by certain technical areas or ports used in a certain building, etc.

Figure 18 shows a NAMES.DAT definition used at the early part of the Fall.

DATA TRANSFER PROBLEM BETWEEN FIELDS

Figure 19 shows our solution to a data transfer problem. Some of our reports showed the user fields by first name and then last name. Channels used by the Trouble Desk had "TROUBLE" as the last name and "DESK" as the first name. "DESK TROUBLE," therefore, appeared on the reports. Information for the Trouble Desk records in these two fields was switched.

ANOTHER DATA TRANSFER PROBLEM IN NAMES.DAT

TEL at one time contained telephone company cable line numbers or dial-up phone numbers for a channel. It was decided that the field BEG, which already contained beginning switch or box numbers, should also contain dial-up phone numbers because the phone numbers marked the beginning of the channel for dial-up ports. TEL would contain only telephone-cable line numbers. Figure 20 shows how the transfer was made. The TEL field was redefined by running another definition of NAMES over NAMES.DAT. All records where TEL contained a phone number were found. BEG was checked to ensure the field was blank. This was a test on the accuracy of the data in the records. A transfer of characters 3 through 8 of TEL was made to BEG. TEL is a 10-character field, and BEG, six characters. "n-nnnn" was transferred to BEG.

LEFT-JUSTIFICATION OF A FIELD

The FIN field needed to be left-justified. The FIN field contains the final concentrator, multiplexer, or switch-port number. Most searches were done on this field, and, therefore, it was extremely important to have consistent coding procedures on this field. Again a temporary dataset definition was placed over NAMES.DAT that redefined FIN (see Fig. 21). Records were found where FIN was not empty but where the first character was blank. The nonblank portion of FIN was stored in a temporary variable and then the temporary variable was stored in FIN. DATATRIEVE left-justifies data that is being entered into a character field.

A DATATRIEVE REPORT ON NAMES.DAT

Figure 22 shows a sample DATATRIEVE report made across NAMES.DAT

CONCLUSIONS

We have had problems with DATATRIEVE on PDP-11 because of the scratch area configuration of DATATRIEVE. Sorts across large datasets are impossible. However, we have used the SRT routine to sort temporary dataset created by DATATRIEVE. The sorted datasets are then used by DATATRIEVE to produce reports. Nested "FOR" loops on a large dataset can take a long time if the FOR's are not searching on indexes. When the processing time for reports became excessive using DATATRIEVE exclusively, we used Fortran programs to create temporary files from our datasets. DATATRIEVE uses these temporary files to make the final reports.

However, DATATRIEVE has proven useful in changing to our database and in providing easily written procedures for non-programmers to enter, modify and report data

ACKNOWLEDGMENTS

Thanks are due to Larry Creel, consultant, for cleaning the original database and consultation on FRAMIS. Thanks are also due to Bob Horning, Los Alamos National Laboratory, for consultation on DATATRIEVE and for Figs. 9, 12, and 18.

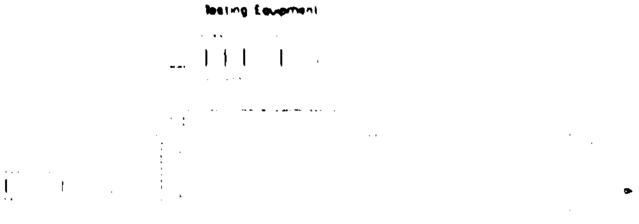


Fig. 1. A POSSIBLE CHANNEL CONFIGURATION.

1. To regain the integrity of the database:
   - A. The old database was an edited file with free form fields.
   - B. Data had been entered without error checking roctines.
   - C. Adequate maintenance of the data was not in place.
2. To gain an interactive database:
   - A. Billing information had to be maintained.
   - B. Technicians on the Trouble Desk needed information about channels at a moment's notice.
   - C. Technicians making cuts for adding new channels or removing old channels needed accurate information.
3. To link to testing equipment for automatic routine checks of channels.

Fig. 2. REASONS FOR THE TRANSFER.

1. Throw away all records that cannot be processed. In the case of our channel information, select only CONTACT, PORT, and DATA CHANNEL records.
2. Throw away all records with blank location codes.
3. Correct consistent typing errors.
4. Throw away duplicates.
5. Standardize coding.
6. Select useful data fields.
7. Match records, e., CONTACT, PORT and DATA CHANNEL records that belong together.
8. Supply missing delds.
9. Separate the data into five datasets based upon the security field.

Fig. 3. TO CLEAN THE DATABASE.

```
                                                              + --
         LOCATION CODE  78B
  CONTACT, SMITH, A.            5521 AT-1          M/S       817--
                      LOC  78AU53/801/TRL
DATA CHANNEL LOW SPEED,   PAIR                                  --
                      LOC  78AU53/801/TRL
                                                              + --
         LOCATION CODE  78B
  CONTACT, SMITH, B.            5521 AT-1          M/S       817--
                      LOC  78BU53/ 40/100
DATA CHANNEL LOW SPEED,   PAIR                               9413--
                      LOC  78BU53/ 40/129
                                                              + --
         LOCATION CODE  81
  CONTACT, TROUBLE DESK  7423                      M/S       254--
                      LOC  81  3/132/123E
CCF PORT                                        O  KCC     42000--
         LL M         LOC  81  3/132/123E
                                                              + --
         LOCATION CODE  82
  CONTACT, SMITH, C.            7510 T-10          M/S       710--
                      LOC  82 U 3/1213/
DATA CHANNEL, SPECIAL               3/1213/108/1                --
                      LOC  82 U 3/1213/108
CCF PORT          95100000                      O  KCC     42061--
         LL P         LOC  82 U 3/1213/108               JACKFIELD
WORK REQUEST                                       LS       4528--
                      LOC  82 U 3/1213/108
```

— At the end of a line indicates a continuing record through the next line

Note the inconsistent coding in some of the fields.

Fig. 4. A PORTION OF THE CHANNEL DA AFTER CLEANING.

```
$ALLOCATE MTA1:
$MOUNT MTA1:  N12345/DENSITY=800 -
$_/BLOCKSIZE=512/FOR
$MCR FLX
FLX MT1:/ZE/DO              - to initialize the tape
FLX MT1:/DO=*.DAT/RS        - to copy disk files to tape
FLX MT1:[*,*]/LI            - to list the datasets on
                              tape
FLX ?
$LOG
```

Fig. 5. TRANSFER OF DATA OFF A 11/780 VAX TO TAPE.

Figures 6, 7, and 8--all portions of the same command file.

```
ALL MT:
MOUNT MT:  N12345/DENS=800/BS=512/FOR
FLX DUO:  [UIC_2]/RS = MT:[UIC_1]*.*
DMO MT:
DEAMT:
```

Note: The PDP-11 is using RSX 11M, Vers. 4.0. There is one tape drive.

Fig. 6. TRANSFER FROM TAPE ONTO A PDP-11.

```
EDT ANAMES.DAT

    SET NOKEYPAD
    CHANGE
    CHGCER                    - change character case from
                                this point in the file to
                                this end of range

    EX                        - exit back to the line editor
    EX                        - exit from EDT saving the
                                edited file
.
.
.

EDT FNAMES.DAT
PIP NNAMES.DAT;*/DE
RUN TAPEFORM                  - a Fortran program to
                                combine three sequential
                                records in the datasets into
                                one record. The program
                                produces five versions of
                                NNAMES.DAT.
```

Fig. 7. MODIFY THE DATA FILES.

```
DTR @ANAMES                  - field definitions
DTR @APATHS
DTR @CNAMES
.
.
.
DTR @UPATHS
DTR @FNAMES
DTR @NFNAMES                 - fills in NAMES dataset
PIP NNAMES.DAT;5/DE
DTR @UNAMES
DTR @ULOCATN                 - fills in the beginning
                               record for a channel in
                               PATHS.  Beginning
                               location for the record is
                               divided up into Tech
                               Area, building and
                               room.
PIP NNAMES.DAT;4/DE
PIP ALTNAM.DAT;*/DE

.
.
DTR @NANAMES
DTR @ALOCATN
PIP NNAMES.DAT;1/DE
PIP ALTNAM.DAT;*/DE
DTR @NPATHS                  - fills in an ending record
                               for the channel in
                               PATHS

BYE
EXIT                         - logs off when procedure
                               finishes
```

Fig. 8. STORE DATASETS INTO DATATRIEVE DATASETS.

```
SET DICTIONARY TOPO
DEFINE DOMAIN ANAMES USING
  ANAMESREC ON [UIC] ANAMES.DAT;
DEFINE RECORD ANAMESREC USING
01 ANAMES-REC.
      15 ID                      PIC IS X(4).
      15 MTLFIL                  PIC IS X(30).
      15 TAG                     USAGE IS INTEGER     PIC IS 9(6)
                                 EDIT-STRING IS Z(8).
      15 TELCO                   PIC IS X(10).
      15 SP                      PIC IS X(2).
      15 STATUS                  PIC IS X.
      15 XMSN                    PIC IS X(3).
      15 FORMAT                  PIC IS X.
      15 SPEED                   PIC IS X(5).
      15 SIGNALLING              PIC IS X(5).
      15 TYPE                    PIC IS X(3).
      15 CONTACT.
            25 LAST                      PIC IS X(15).
            25 FIRST                     PIC IS X(12).
      15 PHONE                   PIC IS X(11)
                                 EDIT-STRING IS X-XXX-XXX-XXXX.
      15 CHARGE-CODE                     PIC IS X(3).
      15 CHARGE.
            25 COSTCTR                   USAGE IS INTEGER
                                         PIC IS 9(4).
            25 PROGCOD                   PIC IS X(4).
      15 LAST-UPDATE                     PIC IS X(9).
      15 CHAN.
            25 FIN                        PIC IS X(6).
            25 START.
                                 35 BEG             PIC IS X(6).
                                 35 LOC             PIC IS X(15).
            25 LOC-CODE                  PIC IS X(10).
;
```

Destination Dataset

Fig. 9. ANAMES.DAT

```
SET DICTIONARY TOPO
DEFINE DOMAIN NNAMES USING
  NNAMESREC ON [UIC] NNAMES.DAT;
DEFINE RECORD NNAMESREC USING
01 NNAMES-REC.
      15 ID                      PIC IS X(4).
      15 MTLFIL                  PIC IS X(30).
      15 TAG                     PIC IS X(6).
      15 TELCO                   PIC IS X(10).
      15 SP                      PIC IS X(2).
      15 STATUS                  PIC IS X.
      15 XMSN                    PIC IS X(3).
      15 FORMAT                  PIC IS X.
      15 SPEED                   PIC IS X(5).
      15 SIGNALLING              PIC IS X.
      15 FILLER                  PIC IS X(4).
      15 TYPE                    PIC IS X(3).
      15 CONTACT.
            25 LAST                      PIC IS X(15).
            25 FIRST                     PIC IS X(12).
      15 PHONE                   PIC IS X(11).
      15 CHARGE-CODE             PIC IS X(3).
      15 CHARGE.
            25 COSTCTR                   PIC IS X(4).
            25 PROGCOD                   PIC IS X(4).
      15 LAST-UPDATE             PIC IS X(9).
      15 NUMALT                  PIC IS X.
      15 ENDING-PORT             PIC IS X(6).
      15 LOCATION-CODE                   PIC IS (10).
      15 BEGINNING-LOC                   PIC IS X(15).
      15 BEGINNING-BOX                   PIC IS X(6).
      15 EXTRA                           PIC IS X(3).
;
```

Transferred data from the 7600.

Fig. 10. NNAMES.DAT

```
SET DICTIONARY TOPO
READY NNAMES
READY ANAMES WRITE
FOR NNAMES
        STORE ANAMES USING BEGIN
              ID=ID
              MTLFIL=MTLFIL
              TAG=TAG
              TELCO=TELCO
              SP=SP
              STATUS=STATUS
              XMSN=XMSN
              FORMAT=FORMAT
              SPEED=SPEED
              SIGNALLING=SIGNALLING
              TYPE=TYPE
              CONTACT.LAST=CONTACT.LAST
              CONTACT.FIRST=CONTACT.FIRST
              PHONE=PHONE
              CHARGE-CODE=CHARGE-CODE
              CHARGE.COSTCTR=CHARGE.COSTCTR
              CHARGE.PROGCOD=CHARGE.PROGCOD
              LAST-UPDATE=LAST-UPDATE
              CHAN.FIN=ENDING-PORT
              CHAN.START.BEG-BEGINNING-BOX
              CHAN.START.LOC=BEGINNING-LOC
              CHAN.LOC-CODE=LOCATION-CODE
              END
FINISH
```

Transfer procedure for NAMES.DAT

Fig. 11. NANAMES.DAT

```
SET DICTIONARY TOPO
DELETE APATHS;
DELETE APATHREC;
DEFINE DOMAIN APATHS USING APATHREC ON [UIC] APATHS.DAT;
DEFINE RECORD APATHREC USING
01 APATH-REC.
     5 TAG              PIC IS 9(6)         USAGE IS INTEGER
                        EDIT-STRING IS Z(8).
     5 SEQ              PIC IS 9(4)         USAGE IS INTEGER
                        EDIT-STRING IS Z(5).
     5 STATUS           PIC IS X.
     5 GRF              PIC IS X(2).
     5 LOCATION.
              15 TA     PIC IS 9(2)         USAGE IS INTEGER.
              15 BLD    PIC IS X(5).
              15 RM     PIC IS X(6).
     5 HWA.
              15 DEV            PIC IS X(6).
              15 BLK            PIC IS X(4).
              14 VIRT           PIC IS X(6).
     5 FILE-SPECS.
     .
     .
     .
     5 LAST-UPDATE      PIC IS X(9).        USAGE IS DATE.
```

Partial listing of destination dataset.

Fig. 12. APATHS.DAT

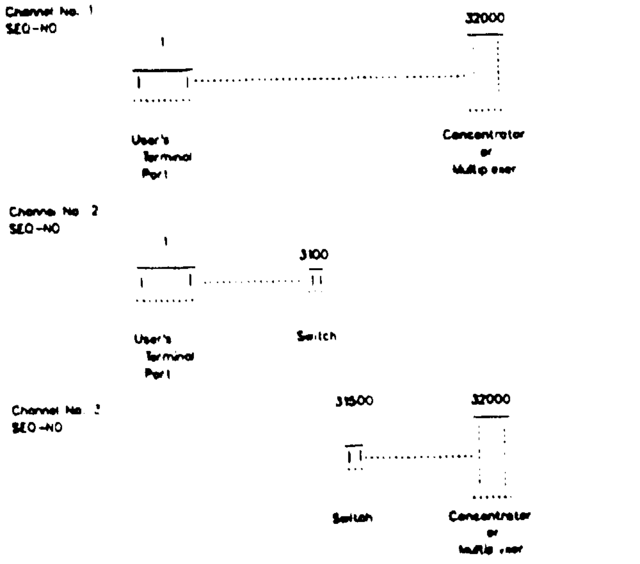


Fig. 13. TYPICAL CHANNEL CONFIGURATION AND CORRESPONDING SEQ-NO ASSIGNMENTS FOR PATHS.DAT

```
SET DICTIONARY TOPO
DELETE ALTNAM;
DELETE THREE-ALTNAMES;
DEFINE DOMAIN ALTNAME USING
  THREE-ALTNAMES ON [UIC] ALTNAM.DAT;
DEFINE RECORD THREE-ALTNAMES USING
01 ALTNAME-REC.
     5 TAG              USAGE IS INTEGER           PIC IS 9(6)
                        EDIT-STRING IS Z(8).
     5 STATUS           PIC IS X.
     5 BEGINNING-BOX    PIC IS X(6).
     5 BEGINNING-LOC    PIC IS X(15).
     5 T1 REDEFINES     BEGINNING-LOC.
              15 EXTRA            PIC IS X.
              15 TA               PIC IS X(2).
              15 SLASH1           PIC IS X.
              15 BLDG             PIC IS X(3).
              15 SLASH2           PIC IS X.
              15 ROOM             PIC IS X(6).
              15 FILLER           PIC IS X.
     5 T2 REDEFINES     BEGINNING-LOC.
              15 EXTRA            PIC IS X(2).
              15 TA               PIC IS X(2).
              15 SLASH3           PIC IS X.
              15 BLDG             PIC IS X(3).
              15 SLASH4           PIC IS X.
              15 ROOM             PIC IS X(6).
     5 T3 REDEFINES     BEGINNING-LOC.
              15 EXTRA            PIC IS X.
              15 TA               PIC IS X(2).
              15 SLASH5           PIC IS X.
              15 BLDG             PIC IS X(4).
              15 SLASH5           PIC IS X.
              15 ROOM             PIC IS X(6).
;
```

Temporary database for data transfer to PATHS databases

Fig. 14. ALTNAM.DAT

```
SET DICTIONARY TOPO
DEFINE FILE FOR ALTNAME;
READY ANAMES SHARED READ
READY ALTNAME WRITE
FOR ANAMES WITH TYPE NOT EQUAL "STA", "COM"
      STORE ALTNAME USING BEGIN
            TAG=TAG
            STATUS=STATUS
            BEGINNING-BOX=CHAN.START.BEG
            BEGINNING-LOC=CHAN.START.LOC
      END
FINISH
READY ALTNAME SHARED READ
DEFINE FILE FOR APATHS ALLOCATION=1375 KEY=TAG(DUP),
   KEY=SEQ;
READY APATHS WRITE
DECLARE SLASH PIC IS X.
SLASH="/"
FOR ALTNAME WITH BEGINNING-BOX EQ " " BEGIN
      IF SLASH1 EQ SLASH AND SLASH2 EQ SLASH THEN
            STORE APATHS USING BEGIN
                  TAG=TAG
                  STATUS=STATUS
                  SEQ=1
                  LOCATION.TA=T1.TA
                        BLD=T1.BLDG
                        RM=T1.ROOM
                  LAST-UPDATE="19-MAY-83"
                  END
      IF SLASH5 EQ SLASH AND SLASH EQ SLASH THEN
            .
            .
      IF SLASH5 EQ SLASH AND SLASH6 EQ SLASH THEN
            .
            .
                  END
FOR ALTNAME WITH BEGINNING-BOX NOT EQUAL " "
      STORE APATHS USING BEGIN
            TAG=TAG
            STATUS=STATUS
            SEQ=31500
            HWA.DEV=BEGINNING-BOX
            LOCATION.TA=3
                  BLD="132"
                  RM="270"
            LAST-UPDATE="19-MAY-83"
            END
FINISH
```

Creation of "beginning" records in PATHS database

Fig. 15. ALOCATN

```
SET DICTIONARY TOPO
READY ANAMES SHARED READ
READY APATHS WRITE
DECLARE SEQ-END          PIC IS 9(4)      USAGE IS INTEGER.
DECLARE XHWA             PIC IS X(6).
FOR ANAMES WITH TYPE NOT EQUAL "STA", "COM" BEGIN
      SEQ-END = 32000
      IF TYPE EQ "PSL" THEN SEQ-END = 31000
      XHWA=FIN
      STORE APATHS USING BEGIN
            TAG=TAG
            STATUS=STATUS
            SEQ=SEQ-END
            GRP="AA"
            HWA.DEV=XHWA
            LAST-UPDATE="TODAY"
            END
      END
FINISH
.
.
.
```

Creation of "ending" records in PATHS database

Fig. 16. NPATHS

1. Elimination of the PATHS.DAT sets.
2. Combination of all security partitions into one dataset.
3. Changes in field lengths.
4. Renaming fields for easier use.
5. Addition of query names for easier access.
6. Regrouping of fields for easier transfers.
7. Addition of new fields.
8. Reshuffling of data within fields.

Fig. 17. SUBSEQUENT CHANGES TO OUR DATABASE.

```
SET DICTIONARY TOPO
DELETE ANAMES;
DELETE ANAMESREC;
DEFINE DOMAIN ANAMES USING ANAMESREC ON [UIC] ANAMES.DAT;
DEFINE RECORD ANAMESREC USING
01 ANAMES-REC.
      15 ID                      PIC IS X(4).
      15 MTL                     PIC IS X([illegible]
   15 TAG                     USAGE IS INTEGER    PIC IS 9(6)
                                 EDIT-STRING IS Z(8).

      15 TELCO.
              25 TEL                  PIC IS X(10).
              25 ALT                  PIC IS X(10).
              25 FRST                 PIC IS X(4).
              25 SEC                  PIC IS X(4).
      15 SP                       PIC IS X(2).
      15 STA                      PIC IS X.
      15 XMS                      PIC IS X(3).
      15 FMT                      PIC IS X.
      15 SPD                      PIC IS X(5).
      15 SIG                      PIC IS X(5).
      15 TYP                      PIC IS X(3).
      15 CONTACT.
              25 LST              PIC IS X(15).
              25 FST              PIC IS X(13).
              25 PHONE              PIC IS 9(11)
                                   EDIT-STRING IS X-XXX-XXX-XXXX
                                  QUERY-NAME IS PHO.

              25 ORG.
                                  35 DIV        PIC IS X(4).
                                  35 GRP        PIC IS X(3).
              25 MAIL          PIC IS X(4).
      15 CHARGE-CODE          PIC IS X(3)       QUERY-NAME IS OC.
      15 CHARGE.
              25 COSTCTR      USAGE IS INTEGER PIC IS 9(4)
                                  QUERY-NAME IS CTR.
              25 PROGOOD      PIC IS X(4)       QUERY-NAME IS PRG.
      15 LAST-UPDATE          USAGE IS DATE  QUERY-NAME IS LU.
      15 CHAN.
              25 FIN              PIC IS X(8).
              25 START.
                                  35 BEG  PIC IS X(8).
                                  35 LOC  PIC IS X(15).
              25 LOC-CODE            PIC IS X(10).
              25 DEV1.
                                  35 DV1              PIC IS X(8).
                                  35 ST1               PIC IS X.
              25 DEV2.
                                  35 DV2              PIC IS X(8).
                                  35 ST2               PIC IS X.
;
DEFINE FILE FOR ANAMES ALLOCATION=535 KEY=TAG KEY=ID
KEY=FIN;
```

Current definition of NAMES.DAT

Fig. 18. ANAMES.DAT

```
READY NAMES SHARED READ
FIND NAMES WITH LST="TROUBLE"
READY NAMES WRITE
FOR CURRENT MODIFY USING BEGIN
      LST="DESK"
      FST="TROUBLE"
      LU="TODAY"
END
```

Fig. 19. SWAPPING DATA BETWEEN FIELDS.

Temporarily redefine TEL as follows:

```
          25 TEL              PIC IS X(10).
          25 TEL1    REDEFINES TEL.
                     35 EXT                 PIC IS X(2).
                     35 NUM                 PIC IS X(6).
                     35 FILLER              PIC IS X(2).
```

Then proceed:

```
FIND PHONESET IN NAMES WITH TEL CONTAINING
  "667-"
FIND CHGSET IN PHONESET WITH BEG EQ "  "
READY NAMES WRITE
FOR CHGSET MODIFY USING BEGIN
      BEG=NUM
      TEL="  "
      LU="TODAY"
      END
```

Fig. 20. PROBLEM: TO TRANSFER LAST FOUR DIGITS AND THE HYPHEN OF ANY PHONE NUMBERS IN TEL TO BEG.

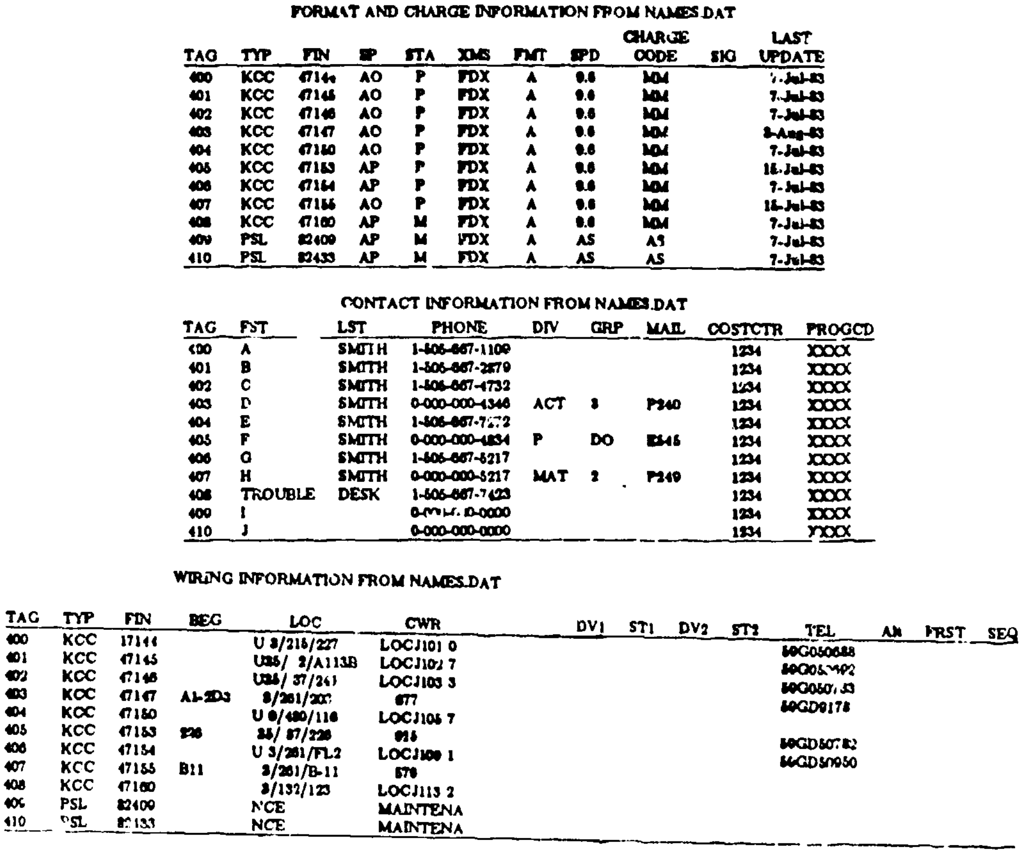
Temporarily redefine FTN as follows:

```
          25 FTN              PIC IS X(8).
          25 FTN1    REDEFINES FTN.
                     35 EMPTY               PIC IS X.
                     35 PRT                 PIC IS X(5).
                     35 FILLER              PIC IS X(2).
```

Then repeat as needed:

```
FIND NAMES WITH EMPTY EQ "  " and PRT NE "  "
DECLARE TEMP PIC X(6).
READY NAMES WRITE
FOR CURRENT BEGIN
      TEMP=PRT
      MODIFY USING BEGIN
            FIN=TEMP
            LU="TODAY"
            END
      END
```

Fig. 21. PROBLEM: TO ELIMINATE LEADING ZEROES IN FIN.



FORMAT AND CHARGE INFORMATION FROM NAMES.DAT

TAG	TYP	FIN	SP	STA	XMS	FMT	SPD	CHARGE CODE	SIG	LAST UPDATE
400	KCC	47144	AO	P	FDX	A	9.6	MM		7-Jul-83
401	KCC	47145	AO	P	FDX	A	9.6	MM		7-Jul-83
402	KCC	47146	AO	P	FDX	A	9.6	MM		7-Jul-83
403	KCC	47147	AO	P	FDX	A	9.6	MM		8-Aug-83
404	KCC	47150	AO	P	FDX	A	9.6	MM		7-Jul-83
405	KCC	47153	AP	P	FDX	A	9.6	MM		15-Jul-83
406	KCC	47154	AP	P	FDX	A	9.6	MM		7-Jul-83
407	KCC	47155	AO	P	FDX	A	9.6	MM		15-Jul-83
408	KCC	47160	AP	M	FDX	A	9.6	MM		7-Jul-83
409	PSL	82409	AP	M	FDX	A	AS	AS		7-Jul-83
410	PSL	82433	AP	M	FDX	A	AS	AS		7-Jul-83

CONTACT INFORMATION FROM NAMES.DAT

TAG	FST	LST	PHONE	DIV	GRP	MAIL	COSTCTR	PROGCD
400	A	SMITH	1-505-667-1109				1234	XXXX
401	B	SMITH	1-505-667-2879				1234	XXXX
402	C	SMITH	1-505-667-4732				1234	XXXX
403	D	SMITH	0-000-000-4346	ACT	3	P240	1234	XXXX
404	E	SMITH	1-505-667-7272				1234	XXXX
405	F	SMITH	0-000-000-4834	P	DO	B545	1234	XXXX
406	G	SMITH	1-505-667-5217				1234	XXXX
407	H	SMITH	0-000-000-5217	MAT	2	P249	1234	XXXX
408	TROUBLE	DESK	1-505-667-7423				1234	XXXX
409	I		0-000-000-0000				1234	XXXX
410	J		0-000-000-0000				1234	XXXX

WIRING INFORMATION FROM NAMES.DAT

TAG	TYP	FIN	BEG	LOC	CWR	DV1	ST1	DV2	ST2	TEL	AN	FRST	SEQ
400	KCC	17144		U 3/215/227	LOCJ101 0					60G050688			
401	KCC	47145		U35/ 2/A113B	LOCJ102 7					60G050692			
402	KCC	47146		U35/ 37/241	LOCJ103 3					60G050733			
403	KCC	47147	A1-2D3	3/261/200	677					60GD9178			
404	KCC	47150		U 0/480/116	LOCJ105 7								
405	KCC	47153	228	25/ 87/228	915					60GD50782			
406	KCC	47154		U 3/261/FL2	LOCJ109 1					60GD50950			
407	KCC	47155	B11	3/261/B-11	579								
408	KCC	47160		3/132/123	LOCJ113 2								
409	PSL	82409		NCE	MAINTENA								
410	PSL	82433		NCE	MAINTENA								

Current sample of database reports

Fig. 22. FORMAT AND CHARGE, CONTACT, WIRING INFORMATION FROM NAMES.DAT

BIBLIOGRAPHY

COMMAND LANGUAGE USER'S GUIDE, VAS/VMS V3.0, Order No. AA-D023C-TE, Digital Equipment Corporation, Maynard, Mass. 01754, May 1982.

DATATRIEVE-11, V2.0, USER'S GUIDE, Order No. AAC742B-TC, Digital Equipment Corporation, Maynard, Mass. 01754, July 1980.

FRAMIS REFERENCE MANUAL; Jones, Stephen E., Ries, Daniel R., Lyles, Leonard, Dittli, Ada L., Johnson, Keith W.; Livermore Computing Systems Document, University of California, Livermore National Laboratory, May 15 1981.

RSX-11M/M-PLUS UTILITIES MANUAL, RSX-11M, Ver. 4.0, RSX-11M-PLUS, Ver. 2.0, Order No. AA-L681A-TC, Digital Equipment Corporation, Maynard, Mass. 01754, November 1981.